A MULTI-AGENT ARCHITECTURE TO SUPPORT ACTIVE FUSION IN A VISUAL SENSOR NETWORK

Federico Castanedo, Jesús García, Miguel A. Patricio and José M. Molina

Applied Artificial Intelligence Group
Computer Science Department
University Carlos III of Madrid, Spain.
{fcastane,jgherrer,mpatrici}@inf.uc3m.es,molina@ia.uc3m.es

ABSTRACT

One of the main characteristics of a visual sensor network environment is the high amount of data generated. In addition, the application of some process, as for example tracking objects, generate a highly noisy output which may potentially produce an inconsistent system output. By inconsistent output we mean highly differences between tracking information provided by the visual sensors. A visual sensor network, with overlapped field of views, could exploit the redundancy between the field of view of each visual sensor to avoid inconsistencies and obtain more accurate results.

In this paper, we present a visual sensor network system with overlapped field of views, modeled as a network of software agents. The communication of each software agent allows the use of feedback information in the visual sensors, called Active Fusion. Results of the software architecture to support Active Fusion scheme in an indoor scenario evaluation are presented.

Index Terms— Visual Sensor Network, Data Fusion, Multi-Agent Systems

1. INTRODUCTION

One of the main characteristics of a visual sensor network environment is the highly amount of data generated. In addition, the application of some process, as for example tracking objects, generate a highly noisy output. This highly noisy information may potentially produce an inconsistent global output of a system. By inconsistent output we mean highly differences between tracking information (positions of the object) provided by the visual sensors. Several factors could provide inconsistent tracking information between visual sensors: occlusions, illumination changes, hardware/software errors.

A visual sensor network with overlapped field of views could exploit the redundancy between the field of view of each visual sensor to avoid inconsistencies and obtain more accurate results. Therefore, a key challenging in visual sensor network contexts is how to get the most relevant information from the environment and fuse it in the most efficient way. However getting the most relevant information from each visual sensor it is not a simple task. Multiple factors could affect the visual sensor information, for example in tracking activities, occlusions of static objects could affect the tracking positions. In this paper, active data fusion is introduced as a technique to get tracking information from the environment and fuse it in the most efficient way.

On the one hand, visual sensor networks (VSNs) are related to spatially distributed multi-sensor environments and cope with distributed computer vision techniques. A distinguish feature of visual sensors, compared with other types such as pressure sensors, microphones, thermometers, is the considerable amount of data generated, which makes mandatory a local processing to deliver only the information represented in a conceptualized level. On the other hand, Multi-Agent systems are defined by the artificial intelligence community as a cooperative network of several intelligent software agents. An intelligent software agent [1] is a computational process which has several characteristics: (1) *"reactivity"* (allowing agents to perceive and respond to a changing environment), (2) *"social ability"* (by which agents interact with other agents) and (3) *"proactiveness"* (through which agents behave in a goal-directed way).

Therefore Multi-Agent VSNs have been introduced in an attempt to achieve more robust, resilient and adaptable computer vision systems endowing them with cognitive faculties: the ability to learn, adaptation, weight alternative solutions, development of new strategies for analysis and interpretation, generalization to new contexts and application domains, and communication with other systems including human beings. These characteristics can be summarized as the ability to reason. In Multi-Agent VSNs, visual sensing is the mechanism or process whereby the system can be influenced by the environment around it. In this paper we present a Multi-Agent VSNs architecture to support active data fusion among each visual sensor information. The information of each visual

This work was supported in part by Projects MADRINET, TEC2005-07186-C03-02, SINPROB, TSI2005-07344-C02-02.

978-1-4244-2665-2/08/$25.00

sensor is managed by a software agent which performs the local signal processing and communicates only the most relevant information (in this paper, tracking positions) to other agents in the system.

One definition of data fusion is [2]: "Data fusion is a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates for observed entities, and to achieve complete and timely assessments of situations and threats, and their significance".

Several works have been published in the area of data fusion. However, the introduction of feedback functionality to adapt the data fusion, known as "adaptive data fusion" are less explored. In [3] the authors provide an overview of adaptive data fusion and sensor management focused on military applications but it can be extended to different applications areas. This work presents a distributed adaptive data fusion, active fusion, applied to the domain of visual sensor networks.

The rest of the paper is organized as follows. Section 2 presents some related works in the area. Then, section 3 briefly introduces the Multi-Agent system architecture. The idea behind the active fusion scheme proposed is presented in section 4. Then, section 5 presents an indoor scenario evaluation using tracking information from a video record of each visual sensor and compared against a ground-truth. Finally, section 6 concludes the paper.

2. RELATED WORKS

The application of Multi-Agent systems in computer vision have been explored in several works. Berge-Cherfaoui [5] proposed a Multi-Agent approach based on the blackboard model. The blackboard model is one of the first Multi-Agent communication models and it is less flexible than current communication models based on FIPA-ACL [6].

Focused on the tracking problem, Marchesotti et.al [7] have been proposed an agent based approach to functionally combine data. Their work is very similar to our proposed system, dealing only with the case of fusing data from two cameras.

In the European project MODEST [8] a Multi-Agent approach based on an information subscription coordination model is used. They deployed four cameras along a bridge in Brussels coordinated through the directory facilitator (DF) of the FIPA platform [9]. Also, they proposed an extension of the Semantic Language (SL) to take into account uncertainty and MPEG-7 descriptors. A different coordination model based on the contract net protocol is used by Graf and Knoll [10]. They distinguish between two types of agents: masters and slaves which are connected using a contract net. Monitorix, another Multi-Agent traffic surveillance system with not overlapped field of views is presented in [11].

The work of [12], which is very similar to our work, proposes an architecture to implement scene understanding algorithms in the visual surveillance domain. The main objective of their work is to obtain a high level description of the events observed by multiple cameras not to fuse the tracking information. As our work, in their architecture each camera is associated to a software agent and the tracking is performed in the ground plane. However, they create one agent per each detected object in the scene in contrast to our proposed architecture which stores object information as an agent belief.

However most of the related works focus on how to solve different visual sensor problems, there are less works focusing on how to build a software architecture which allows an intelligent visual sensor network in which the agents take active part in the fusion process.

3. MULTI-AGENT SYSTEM ARCHITECTURE

Our approach to support a visual sensor network is based on a Multi-Agent system where each visual sensor information is processed by an intelligent agent. In the last years, many Multi-Agent languages and frameworks have been developed [4]. We choose the open source framework Jadex [14], which is FIPA compliant and it is gained acceptance by the Multi-Agent community. Jadex [14] is a Belief-Desire-Intention (BDI) Multi-Agent model which is FIPA compliant [13]. The BDI model provides a way to conceptualize the system and structure its design [15]. The architecture is built by using different types of agents and it is described more in detail in [15, 16]. Each agent has its own responsibilities and cooperate each other in order to make a coherent distributed data fusion. In this paper we briefly describe two different types of agents: (1) Sensor agent and (2) Fusion agent.

3.1. Sensor agent

Each sensor agent S_i acquires images $I(i,j)$ at a certain frame rate, V_i. The detected target of interest O_j is represented with a track vector $\hat{x}_j^i[n]$, containing the numerical description of their attributes and state: location, velocity, dimensions, and associated error covariance matrix, $R_j^i[n]$ (see Fig. 2). In an internal process, target location and tracking are expressed in pixel coordinates, which are local to each i-th camera agent view, S_i, and n is the temporal index associated with time t_n. For more details of this local video tracking process, see [19, 20] Therefore, each sensor known its own detected objects location in local coordinates.

A precalibration process based on the human perception of 3D structures from 2D information, using the pinhole model [21] is performed. Then, these local estimates (or track vectors) are projected to common global coordinates using the mathematical relation between the space points and their equivalents in the camera image. After the projection is performed the information from the detected objects of each sensor agent is send to the fusion agent. A specific parameter (*Looking Interval* in milliseconds) sets the frequency when sending the

detected objects under the field of view.

An agent's *beliefs* correspond to the information the agent acquires from the environment and the other agents. It represents the knowledge of the state of the world. Sensor agents have the following *beliefs*:

1. Environment Knowledge: the knowledge about the detected objects in their field of view.
2. Environment Updating Frequency: an internal parameter which specifies the time in milliseconds of the updating frequency from the detected objects in the environment. The agent must have the necessary balance between the computational effort spent in obtaining visual information and the execution of other activities. This balance is established in the system by using an updating frequency belief value.
3. Communication Frequency: that kind of agent sends the information to the fusion agent periodically. This parameter is established in the system by using a specific belief. Of course, there must be a balance in this parameter to avoid network congestion and ensure the communication.
4. Fusion Agent: each sensor agent know their respective fusion agent name and address (FIPA Agent name).

There are also some configuration parameters:

1. Foreground Algorithm: each agent knows the foreground algorithm used to detect moving objects. This algorithm must provide a list of blobs which are found in a frame. Containing information about the position and size of each blob. It could be possible to use different foreground algorithms in each agent.
2. Camera Type: the knowledge about the type of camera which is being used. Also it could be an input video file.

Desires capture the motivation of the agents. A desire represents the state of affairs that the agent would like to bring about. The sensor agents have the following *desires*:

1. Tracking: this desire regards with the tracking intentions which is performed continuously.
2. Looking: the looking desire allows the agent to observe the current tracks in the environment.
3. Communication: These type of agent has communication desires.

Intentions are the basic steps chosen by the agent to achieve its *Desires* and represent the desires an agent has committed to achieve. *Intentions* constrain the reasoning an agent is required to perform in order to select the action that has to be performed. Surveillance-sensor agents have the following *Intentions*:

1. Tracking: Each sensor agent S_i acquires images $I(x, y)$ at a certain frame rate, V_i. The internal tracking process provides for each object X_{T_j}, an associated track vector of features $\hat{X}_{T_j}^{S_i}[n]$, containing the numeric description of their features and state: location, velocity, dimensions, etc. and associated error covariance matrix, $\hat{P}_{T_j}^{S_i}[n]$.
2. To Look the Environment: This intention performs the observation of the current objects in the environment at time t.
3. New Track Information: The intention to communicate the information about new tracks in the environment to their respective fusion agent.
4. Update Track Information: All the sensor agents can communicate to their respective fusion agent information about the new track features.
5. Delete Track Information: This intention allows the sensor agent to communicate information about a disappeared object.

The previous intentions are the basic steps that allow the architecture to obtain a continuity in the tracking along the field of view of the cameras involved in the distributed network.

3.2. Fusion agent

The fusion agent receives tracks information from the sensor agents through a TCP/IP network using FIPA ACL messages performs the fusion of the data received. . The most important fusion agent parameters involved in the fusion process are:

1. *Temporal Difference:* It is a value in milliseconds which is used to discriminate when the measurements are from different tracks.
2. *Spatial Difference:* It is a parameter in centimeters which specify a threshold used to discard a track due to a spatial inconsistency regarding the others tracks.
3. *Feedback Frequency:* It is a value in milliseconds, used in the active fusion which indicates the frequency of feedback messages.
4. *Fusion Frequency:* This parameter sets the frequency in the fusion process. Every *fusion frequency* milliseconds the fusion process is performed by the fusion agent.

Some fusion agent configuration parameters, are:

1. *Fusion Type:* It indicates the fusion type: active fusion (with feedback) or passive fusion (without feedback).
2. *Fusion Algorithm:* It establishes the fusion algorithm used.

The data fusion process used is based on [16], and it involves: (1) Consistency checking and (2) track fusion between consistent tracks.

(1) Consistency checking: Tracking information provided by each visual sensor should be coherent, therefore different visual sensors should not show big differences in the spatial information about the same object. Consistency checking discards inconsistent tracks in the visual sensor network. It is applied across all received tracks by calculating the Mahalanobis Distance (MD) between all sensor agent pairs (S_i, S_j) to track the features of all transformed vectors:

$$MD_{S_i,S_j} = \left(\hat{x}_l^i[n] - \hat{x}_l^j[n]\right)^t \left(R_l^i[n] + R_l^j[n]\right)^{-1} \left(\hat{x}_l^i[n] - \hat{x}_l^j[n]\right) \leq \lambda$$

If the MD exceeds the λ threshold, the track pair is labeled as inconsistent, indicating that one member of the pair should be discarded from the fusion process. And the sensor agent is warned by the $FeedbackMessage$ message.

(2) Track fusion between consistent tracks: Once consistent tracks have been selected, the data fusion is performed according to each track's reliability. We take a simple federated fusion approach [17], based on weighting each source of information according to the covariance error matrix, modified by an additional score function assessing the confidence level assigned to the tracking process [18]. For each j-th object being tracked in the visual sensor network by i-th camera, the combination is given by

$$\left(R_j^F\right)^{-1} = \sum_{i \in C} \left(\alpha_j^i R_j^i\right)^{-1} ; \; \hat{x}_j^F = R_j^F \sum_{i \in C} \left(\alpha_j^i R_j^i\right)^{-1} \hat{x}_j^i$$

The level of confidence for each consistent camera and for each common target is based on the inverse covariance value of each sensor and target multiplied by the heuristic score function α_j^i. The score function $\alpha_j^i \in [1, \inf)$ is a scalar characterizing the performance of the i-th sensor's camera based on a combination of image tracking performance metrics (combination of color, spatial regularity, shape uniformity, motion stability, etc.).

4. ACTIVE FUSION

The presence of multiple data sources and fusion nodes provides many possibilities in a visual sensor network architecture [22]. In the data fusion literature three different types of distributed schemes are widely adopted: (1) Passive fusion, (2) Active fusion and (3) Peer to Peer fusion. In this paper, we focus on the active fusion scheme, an illustrative figure of this scheme is shown in figure 1.

The idea behind active data fusion scheme is to provide feedback information to each sensor agent involved in the fusion process. This feedback information allows each sensor agent reasoning about the quality of the information which is being sent to the fusion agent regarding the other overlapped sensors. As each sensor agent is autonomous, it can decide

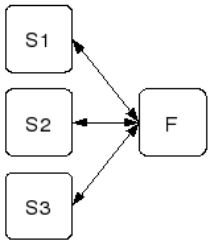

Fig. 1. Active Fusion Architecture

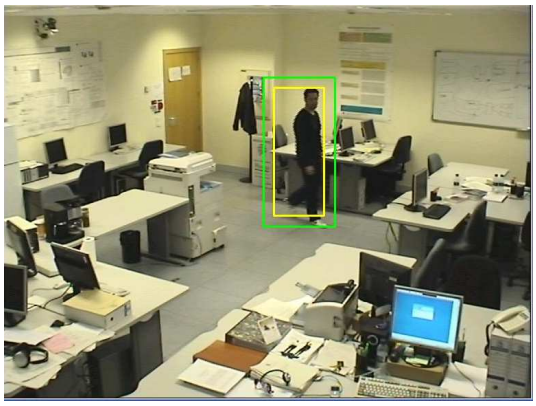

Fig. 2. An example of a detected object. Yellow bounding-box show tracking values.

about the inconsistencies in the information and correct them before they are sent to the fusion agent. This process involves an alignment of the information in order to obtain a coherence of the fusion process by means of a cooperative mechanism. Therefore, Active Fusion implies that each sensor agent is able to manage its local fusion process accordingly to external information, performing actions such as: correct values, delete objects and change parameters of projections.

In this type of scheme the fusion process should deal with data incest. Data incest refers to the inadvertent multiple use of raw measurements several times as though they were independent which can lead to biases in estimates and over confidence in their accuracy. Data incest risk with this scheme of active fusion is moderated, as feedback information can be used first to correct local tracks, which are used later to obtain the fused result in next fusion iteration.

In the case of Active Fusion, when sensor agents receive the feedback messages, a decision process is carried out in order to correct the possible deviations. Therefore, active fusion scheme is an extension of the classical passive fusion scheme which exploits the agents communication ability.

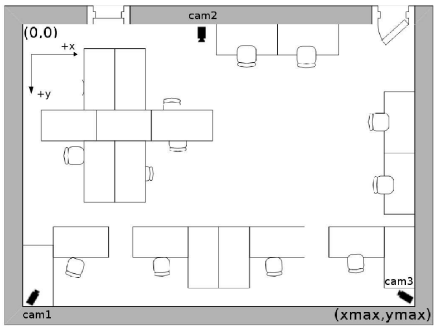

Fig. 3. Experimental environment

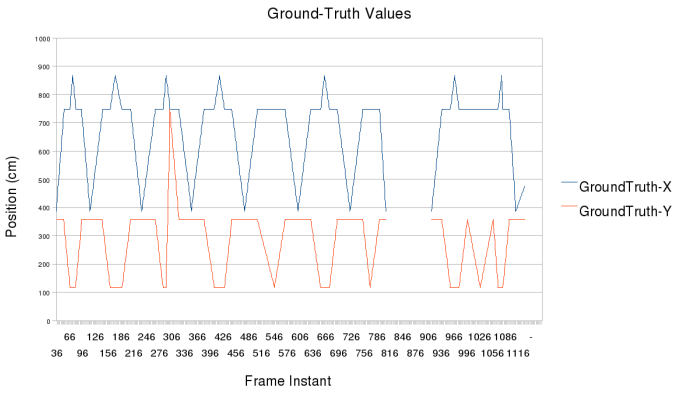

Fig. 4. Ground-truth values of the scenario evaluation

The main sensor agent parameters involved in the Active Fusion process are:

1. *Feedback Threshold:* It establishes the frequency of receiving the fusion feedback.
2. *Spatial Difference:* It is a value in centimeters which specifies a threshold to detect inconsistent measurements with respect to the fused values.

5. SCENARIO EVALUATION

In this section, a scenario evaluation of the proposed architecture is presented. We considered a real scenario where 3 Sony EVI-100 cameras are deployed in a room of 660x800 centimeters (see Fig. 3).

A synchronized video stream from the three cameras was recorded. Each video file has 48 seconds of length and was grabbed at 25 frames per second (1200 frames on each video sequence). The input frames of each video file was processed by a sensor agent and each of them are running in different machines and connected by a TCP/IP network. The input video frames were processed at an average of 5 frames per second. The tracking information was evaluated against ground truth values.

Since the movement was performed in a predefined way, following specific points of the room, the ground truth values were directly obtained and stored. Therefore, we know the global values in the real world of some specific frames and the position in the others frames were interpolated. An illustration of the ground-truth values is presented in figure 4.

The following experiments were performed. architecture was tested in three different situations: using the information of only one visual sensor (sensor1, sensor2 and sensor3).

5.1. Passive Fusion

The tracking positions in global coordinates (called passive fusion) performed by: only visual sensor 1, only visual sensor 2 visual, only sensor 3 and from all of them compared with the ground-truth positions are shown in figures 5, 7, 9 and 11. Also, the mean absolute error of the passive tracked positions against the ground-truth is shown in table 1.

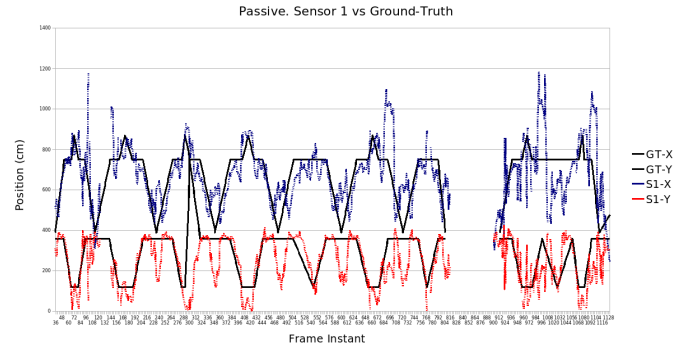

Fig. 5. Visual sensor 1 tracking values in global coordinates compared with the ground-truth in the passive mode.

5.2. Active Fusion

Feedback information provided by the fusion agents allows sensor agents reasoning about the information being sending. In these experiments the ground-truth values were used as feedback information, therefore we simulate a perfect feedback information.

Active fusion tracking positions in global coordinates from: only visual sensor 1, only visual sensor 2, only visual sensor 3 and from all of them compared with the ground-truth positions are shown in figures 6, 8, 10 and 12. The active fusion mean absolute error (table 1) outperforms the error given by the passive fusion scheme. The variance of the positions showed in the plots is given because the sensor agents correct the information before sending to the fusion agent but do not change the tracking state of the object.

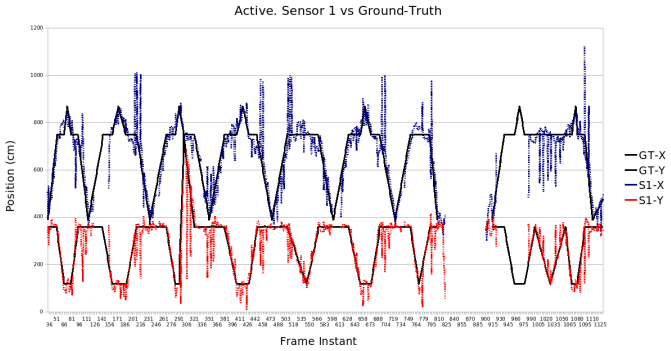

Fig. 6. Visual sensor 1 tracking values in global coordinates compared with the ground-truth in the active mode. The ground-truth was used as feedback information.

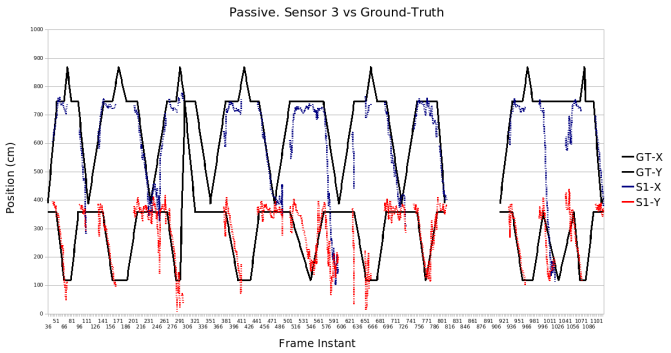

Fig. 9. Visual sensor 3 tracking values in global coordinates compared with the ground-truth in the passive mode.

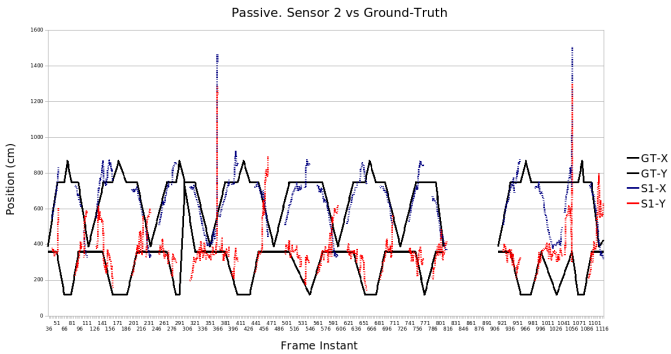

Fig. 7. Visual sensor 2 tracking values in global coordinates compared with the ground-truth in the passive mode.

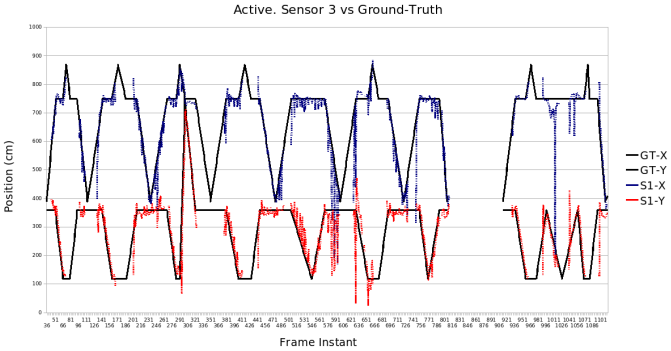

Fig. 10. Visual sensor 3 tracking values in global coordinates compared with the ground-truth in the active mode.

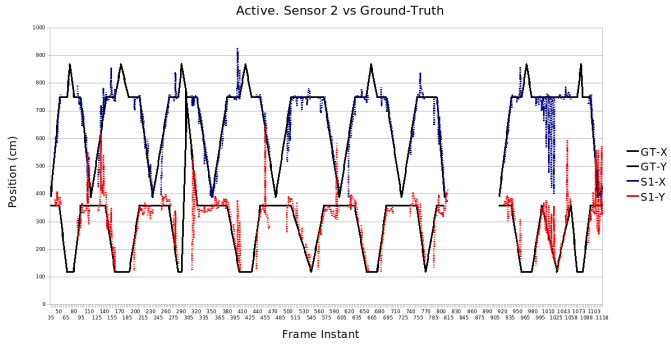

Fig. 8. Visual sensor 2 tracking values in global coordinates compared with the ground-truth in the active mode.

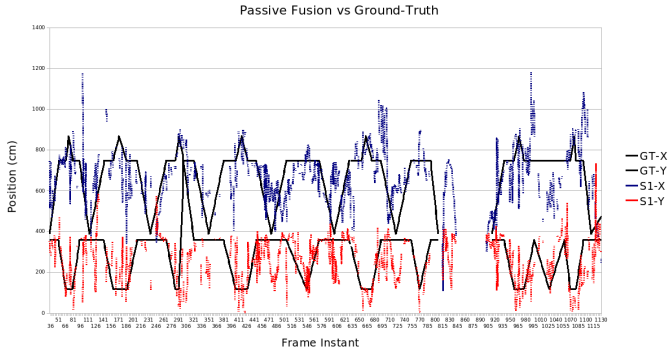

Fig. 11. Fused tracking values in global coordinates compared with the ground-truth in the passive mode.

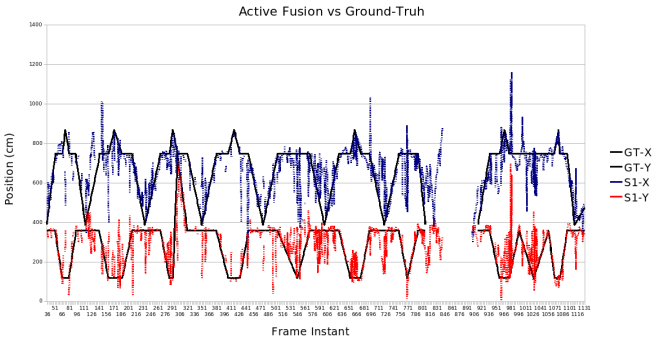

Fig. 12. Fused tracking values in global coordinates compared with the ground-truth in the active mode.

Table 1. Mean absolute error between ground truth and tracking position (centimeters). Passive vs Active.

	S1	S2	S3	S1-S2-S3
Passive error (X)	89.28	69.39	67.17	86,04
Active error (X)	43.15	25.01	22.36	56,23
Passive error (Y)	70.37	85.2	46.48	71.85
Active error (Y)	23.28	33.5	26.42	40,03

6. CONCLUSIONS AND FUTURE WORK

In this paper a visual sensor network design using the Multi-Agent paradigm is presented. The idea behind this architecture it is gained acceptance. An active fusion architecture is presented and experimental results in an indoor scenario using the visual sensor network multi-agent architecture are shown.

The main advantages obtained in a visual sensor network with this architecture are: (1) To build an open architecture which is easy to scale. We could easily add new agents (with different or same goals) in the Multi-Agent system. (2) A standard based architecture, which would allow us to interoperate with third part developments and (3) the explicit use of feedback information to cooperatively improve the fusion process. Experimental results showing the improvement of use active fusion are presented.

The proposed architecture support testing various fusion algorithms over the same dataset. The ground truth values of the dataset allows an accuracy measurement in order to test different fusion algorithms. The impact of the proposed architecture has been analyzed by assessing two data fusion alternatives over the same dataset representing a real scenario with three visual sensors.

7. REFERENCES

[1] M. Wooldridge and N. Jennings, "Intelligent agents: Theory and practice," *The knowledge Engineering Review*, 1995.

[2] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House Inc, Norwood, Massachussets, U.S, 1990.

[3] A.R. Benaskeur and F. Rhéaume- Adaptive data fusion and sensor management for military applications. *Aerospace Science and Technology, vol. 11, no. 4, pp. 327–228, 2007.*

[4] V. Mascardi, D. Demergasso and D. Ancona. "Languages for Programming BDI-style Agents: an Overview," *WOA*, pp. 9–15, 2005.

[5] V. Berge-Cherfaoui and B. Vachon, "A multi-agent approach of the multi-sensor fusion," *Fifth International Conference on Robots in Unstructured Environments*, pp. 1264-1259, 1991.

[6] FIPA, ACL "Message Structure Specification ". *Foundation for Intelligent Physical Agents,* 2000.

[7] L. Marchesotti, L. Piva and C. Regazzoni, "An agent-based approach for tracking people in indoor complex environments." *12th International Conference on Image Analysis and Processing, 2003.*. pp. 99–102, 2003.

[8] Multimedia Object Descriptions Extaction from Surveillance Types. http://www.tele.ucl.ac.be/PROJECTS/MODEST/.

[9] L. Botelho, R. Lopes, M. Sequeira, P. Almeida, and S. Martins, "Inter-agent communication in a FIPA compliant intelligent distributed dynamic-information system," *Proc. 5 thInternational Conference on Information Systems Analysis and Synthesis (ISAS99)*, 1999.

[10] T. Graf and A. Knoll, "A Multi-Agent System Architecture for Distributed Computer Vision," *International Journal on Artificial Intelligence Tools*, vol. 9, no. 2, pp. 305–319, 2000.

[11] B. Abreu, L. Botelho, A. Cavallaro, D. Douxchamps, T. Ebrahimi, P. Figueiredo, B. Macq, B. Mory, L. Nunes, J. Orri, *et al.*, "Video-based multi-agent traffic surveillance system," *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pp. 457–462, 2000.

[12] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. A. Jones, "A multi-agent framework for visual surveillance," in *ICIAP '99: Proceedings of the 10th International Conference on Image Analysis and Processing*, (Washington, DC, USA), p. 1104, IEEE Computer Society, 1999.

[13] P.D. O'Brien and R.C Nicol, "FIPA—Towards a Standard for Software Agents" *BT Technology Journal*, vol. 16, no. 3, pp. 51–59, 1998.

[14] A. Pokahr, L. Braubach and W. Lamersdorf, "Jadex: Implementing a BDI Infraestructure for JADE Agents," *Search of Innovation (Special Issue on JADE)*, vol. 3 pp. 76–85, September, 2003.

[15] F. Castanedo, M.A. Patricio, J. Garcia and J.M. Molina. "Extending surveillance systems capabilities using BDI cooperative sensor agents," *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pp. 131–138, 2006.

[16] F. Castanedo, M.A. Patricio, J. Garcia and J.M. Molina. "Robust data fusion in a visual sensor multi-agent architecture," *10th International Conference on Information Fusion*, pp. 1–7, 2007.

[17] D.L. Hall and J. Llinas. "Handbook of MultiSensor Data Fusion". *CRC Press. Boca Raton*, 2001.

[18] G.L. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis. "Active video- based surveillance system: the low-level image and video processing techniques needed for implementation". *Signal Processing Magazine*, IEEE, 22(2):25–37, 2005.

[19] J.A. Besada, J. Garcia, J. Portillo, J.M. Molina, A. Varona, and G. Gonzalez. "Airport surface surveillance based on video images". *IEEE Transactionbs on Aerospace and Electronic Systems*, vol. 41 no. 3, pp. 1075—1082, July 2005.

[20] O. Perez, M.A. Patricio, J. Garcia and J.M. Molina. "Improving the segmentation stage of a pedestrian tracking video-based system by means of evolution strategies". *In EvoWorkshops*, pp. 438–449, 2006.

[21] R. Tsai- "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses". *IEEE Journal of Robotics and Automation*, vol. 3, no. 4 pp. 323–244, 1987.

[22] M.E. Liggens, C.Y. Chong, I. Kadar, M.G. Alford, V. Vannicola and S. Thomopoulos. "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85 pp. 95–107, 1997.